RIPHAH INTERNATIONAL UNIVERSITY

Software Construction & Development
Fall 2022

Delico - A Cafe Management System

Project Team

Name of Students	Sap ID	Program	Valid Email Address
Mariam Farooqi	26849	BSSE	26849@students.riphah.edu.pk
Javairia Armakoon	24064	BSSE	24064@students.riphah.edu.pk
Zunaira Shahid	22656	BSSE	22656@students.riphah.edu.pk
Safa Sohail	13254	BSSE	13254@students.riphah.edu.pk

Date of Submission
1/11/2022

Artifact #1
Project Proposal

Introduction of the Project

Project Title: Delico – A Cafe Management System

Introduction:

The domain of our project is Cafe Management System which is based on a concept to maintain orders and management of particular items. It is the system that provides the customer facility to buy food items directly from a seller through the internet without using an intermediary provider. It entails going online, landing on a seller's website, making a purchase, and coordinating delivery. Furthermore, it provides customer servicesthat when a customer finds a food item, they will land on a website that allows them to add multiple items and adjust quantities to cart. The customer can pay for the food item using a credit or debit card online or at the time of delivery. The food items can be delivered to customer once payment has been accepted.

On the other hand, manager is working to keep the information up to date in the system, including processes like updating, deleting, and adding item records and customer order records. This system helps to simply ease out their day-to-day managerial task

Problem Statement:

When cities became more developed and societies became vaster, it becomes difficult for customer to go to market to bring food. Mainly cars are not available to every specific person and walking for about 2 kilometers is not an easy task. By walking about 20 to 30 minutes and then going to market and selecting things is a waste of time. Sometimes some items are not available or they are out of stock. Occasionally no one is available to fetch things, we are far away from restaurants and due to some problems like insufficient fuel, tire burst, vehicle bad condition etc., we are not able to get what we want. Due to this customer gets frustrated. Apart from this, at times guests come at a sudden and there is nothing to make at home so we have to rush far away to markets to buy things. These difficulties make customer stressful.

Proposed Solution:

We are going to develop a system named as Delico – A Cafe Management System that would allow customer to order online. Through this system customer would be able to order food while sitting at home. This will save much time and customer will be out of the tensions of finding someone to bring those items. Customer will not be facingdifficulties of travelling and then wasting time for selecting items. By just one click customer can order food and within some time items will be delivered. Delico – A Cafe Management System will ensure that there are special offers and sale on for customers to save their money. Altogether ensuring customer satisfaction.

Scope of the Project:

Delico- A cafe management that accommodates online ordering of food by customers also ensuring that customer is thoroughly satisfied. Apart from this, the central system functionalities of this system comprise of bill generation, maintaining staff records, and managing orders. The order management involves the deleting of orders, removing and adding food. All the orders made are saved in system and tied to the significant project tasks, with testing and implementation that consume marginally more time than design.

Modules:

Manage Item
Manage Category
Manage Cart
Manage Staff
Manage Offers
Manage Order
Manage Stock
Customize Settings

Modules Description:

Manage Items

This module provides add, view, delete, and update cafe related functionalities to the manager.

- Add items
- View items
- Delete items
- Update items

Manage Category

This module provides add, delete and update functionalities to the manager.

- Add category
- Delete category
- Update Category
- View Category

Manage cart

This module provides add, delete and view functionalities to the customer.

- View Items
- Delete Items
- Add Items

Manage Staff

This module provides add, delete and update functionalities to the manager.

- Add staff
- Delete staff
- Update staff
- View Staff

Manage Offers

This module provides add, delete and update functionalities to the manager.

- Add Offer
- Delete Offer
- Update Offer
- View Offer

Customize Settings

This module provides update and view functionalities to the customer.

- View Settings
- Update Settings

Manage Order

This module provides add, delete and view functionalities to the manager.

- Add Order
- Delete Order
- View Order

Manage Stock

This module provides add, delete, view and update functionalities to the manager.

- Add Stock
- Delete Stock
- Update Stock
- View Stock

Artifact #2
Usecase Diagram

Use Case Diagram:

Actors:

- Manager
- Customer
- System

Use Cases:

- Register
- Login
- Place Order
- Manage Cart
- Make Payment
- Payment via cash
- Payment via credit card
- Manage Item
- Add Items
- Manage Staff
- Add Staff
- Manage Category
- Add Category
- Manage Offers
- Add Offers
- Manage Order
- Manage Stock
- Add Stock
- Customize Settings
- Generate Bill

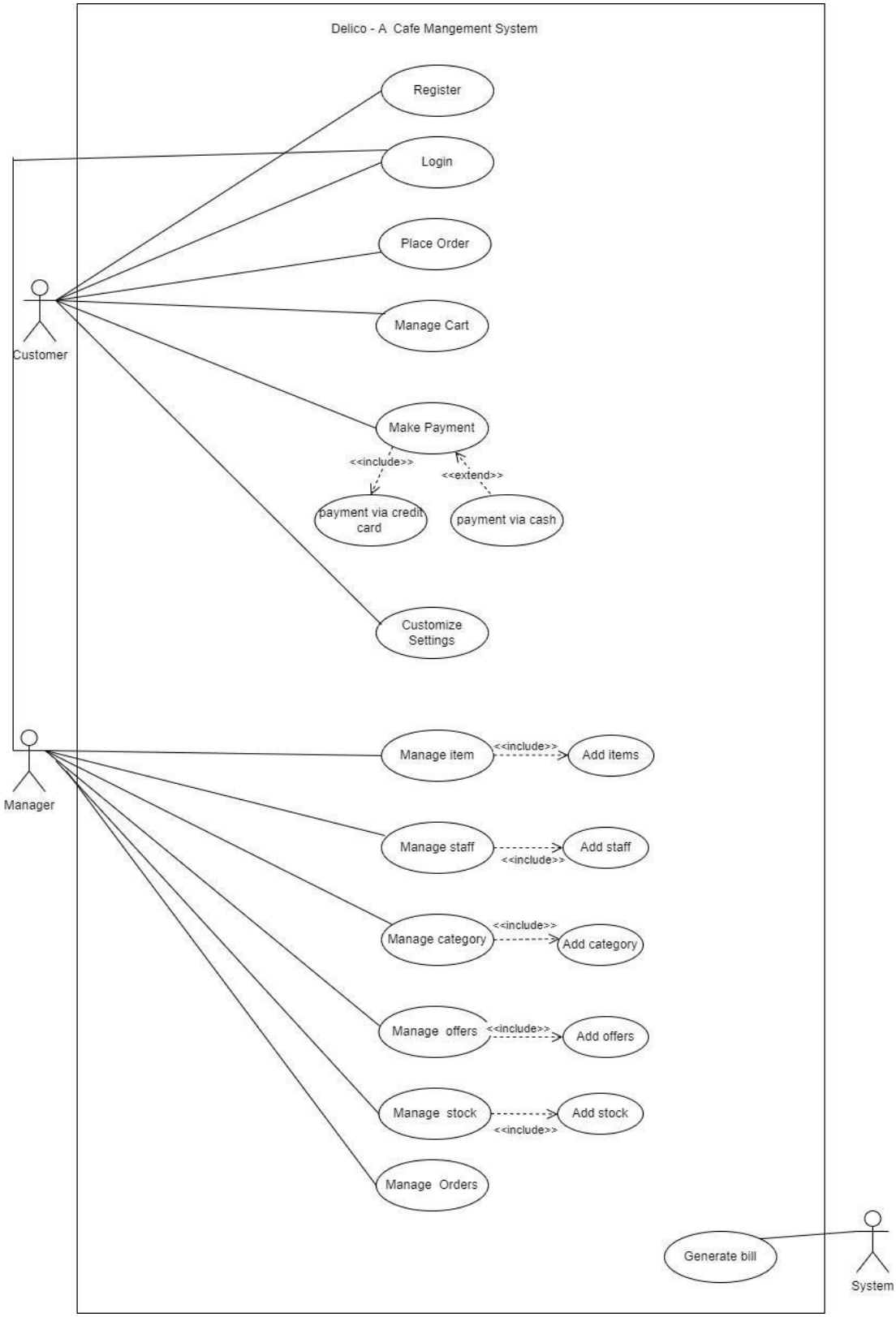


Fig 2 Use case Diagram

Artifact #3
Fully Dressed

1. ID: UC- 01	
Use case: Register	
Brief Description**:** This use case allows the customer to register in the system to access specific functions such as place order. To register into system customer, have to enter their valid username, mobile number, password and address.	
Primary actors: Customer	
Preconditions: Customer should enter valid credentials.	
Post condition: Customer has registered successfully and system displays message of Account Created Successfully.	
Main Flow:	
Actor: 1. Customer starts registration process. 3. Enters all the required information i.e. username, mobile number, password and address. 4.Customer submits their request for registration t system.	System: 2. System displays registration page which includes all the necessary information such as username, mobile number, password and address. 5. System displays message of Account Created Successfully. 6. System save account details of customer.
Alternative flow:	
Actor: 3a) Missing required information.	System: 3a) 1. System displays “Fill the Missing Credentials” message. 3a) 2. Use case resumes at main flow step 3.
Frequency of use:	

High
Technology and Variation: Java (Net beans IDE)
Special Requirements: **Usability:** System is easy to understand for customers. **Performance:** System responds to all customer clicks within 1 second.

2. ID: UC- 02	
Use case: Login	
Brief Description**:** This use case allows the user to log in to the system to access specific functions depending on their roles. To login into system user have to enter their valid username and password. Upon successful login system will display user's relevant home page.	
Primary actors: Customer, Manager (User)	
Preconditions: 1. Customer should be registered and must have valid account. 2. Manager should have valid account.	
Post condition: System will display a message Welcome to Delico Cafe.	
Main Flow:	
<u>Actor:</u> 1. User starts login process. 3. User enters valid username and password to access account. 4. User submits their request for login to system.	<u>System:</u> 2. System displays login page so that user enters their username and password. 5. System displays message Welcome to Delico Cafe.

6. User submits their request to view their relevant homepage.	7. System displays homepage according to user status (manager, customer).
Alternative flow:	
Actor: 3a) Invalid username or password.	System: 3a) 1. System displays "Username and Password incorrect "message. 3a) 2. System prompts to reenter Username or Password. 3a) 3. Use case resumes at main flow step 3.
Frequency of use: High	
Technology and Variation: Java (Net beans IDE)	
Special Requirements: **Usability:** System is easy to understand for customers. **Performance:** System responds to all customer clicks within 1 second. **Robustness:** In case any incomplete information or any invalid input is entered system displays error message on screen within 1 second.	

3. ID: UC- 03
Use case: Place Order
Brief Description**:** Customer accesses the system and chooses the required item and when customers get the items of their choice, they confirm the order of selected item.

Primary actors: Customer	
Preconditions: 1. The record of items exists to view it. 2. Customer should be login into system.	
Post condition: Order of items has been placed successfully.	
Main Flow:	
Actor: 1. Customer starts placing order process. 3. Customer chooses items from menu list. 4. Customer adds items in cart. 5. Customer submits their request for generate bill.	System: 2. System displays list of menu. 6. System displays bill to customer. 7. System saves the information of customer.
Alternative flow:	
Actor: None	System: None
Frequency of use: High	
Technology and Variation: Java (Net beans IDE)	
Special Requirements: **Performance:** System responds to all customer clicks within 1 second.	

4. ID: UC- 04
Use case: Manage cart

Brief Description**:** In this use case customer is preparing for an ordering process with adding items to or removing items from the cart.	
Primary actors: Customer	
Preconditions: Customer must select any item.	
Post condition: 1. Customer will successfully add or remove item from cart. 2. The total bill will be recalculated if customer changes the number of items.	
Main Flow:	
Actor: 1. Customer adds to or remove items from cart.	System: 2.System allows customer to add and remove item from cart and displays specific items being added to or removed from cart.
Alternative flow:	
Actor: None	System: None
Frequency of use: High	
Technology and Variation: Java (Net beans IDE)	
Special Requirements: **Learn ability**: User interfaces are easy so that customers can understand system functionality. **Performance:** System responds to all customer clicks within 1 second.	

5. ID: UC- 05	
Use case: Make payment	
Brief Description**:** In this use case customer makes payment of the item they have ordered by choosing payment options provided to them such as payment by credit card and cash.	
Primary actors: Customer	
Preconditions: Customer has placed order for the item.	
Post condition: System displays message Payment done successfully.	
Main Flow:	
Actor: 1. Customer starts payment process. 4. Customer chooses option such as payment by credit card and cash.	System: 2. System displays interface of payment. 3. System prompts user to choose payment method.
Alternative flow:	
Actor: 4a). Unchecked payment mode. 4b). Paying by credit: include Payment by credit card. 4b) Paying by cash: 1. Extend Payment by cash.	System: 4a) 1. Display "Please choose payment mode" error message. 4a) 2. Prompts customer to choose payment mode. 4a) 3. Use case resumes at main flow step 4.
Frequency of use: High	
Technology and Variation: Java (Net beans IDE)	

Special Requirements: **User Interface:** System shall have a user-friendly interface so that customer can understand it easily.

6. ID: UC- 06	
Use case: Payment by credit card	
Brief Description**:** In this use case customer makes payment of the item they have ordered by selecting payment mode such as credit card.	
Primary actors: Customer	
Preconditions: 1. Customer has placed order for the product. 2. Customer must have valid credit card.	
Post condition: System displays message Payment done successfully.	
Main Flow:	
<u>Actor:</u> 1. Customer selects credit card option 3. Customer enters credit card information. 4. Customer submits credit card information to system.	<u>System:</u> 2. System displays interface for credit card information and total bill. 5. System validates credit card information entered by user. 6. System displays message “Payment done successfully”
Alternative flow:	

Actor:	System:
3a). Incorrect credit card information. 3b). Missing credit card information.	3a) 1. System displays error message “Input right value” 3a) 2. System prompts customer to reenter information. 3a) 3. Use case resumes at main flow step 3. 3b) 1. System displays error message “Please fill in the required spaces” 3b) 3. Use case resumes at main flow step 3.
Frequency of use: High	
Technology and Variation: Java (Net beans IDE)	
Special Requirements: **Learn ability:** User interfaces will be easy for so that customer can understand system functionality. **Performance:** 1**.** When customer searches any item system will take 2 sec to display relevant product image. 2. System will respond to all clicks of customer within 5 seconds.	

7. ID: UC- 07
Use case: Payment by cash on delivery
Brief Description**:** In this use case customer makes payment of the item they have ordered, manually by selecting payment mode such as cash on delivery.

Primary actors: Customer	
Preconditions: Customer has placed order for the product.	
Post condition: System displays message Payment done successfully.	
Main Flow:	
Actor: 1. Customer selects cash on delivery option.	System: 2. System displays interface for credit card information and total bill. 3. Displays message “Pay cash to courier when order has been received.”
Alternative flow:	
Actor: None	System: None
Frequency of use: High	
Technology and Variation: Java (Net beans IDE)	
Special Requirements: **Learn ability:** User interfaces will be easy for so that they can understand system functionality.	

8. ID: UC- 08
Use case: Customize Settings
Brief Description**:** In this use case customer will be able to change their profile settings such as

username, mobile number, password and address.	
Primary actors: Customer	
Preconditions: Customer should have valid account.	
Post condition: System displays message Profile settings changed successfully.	
Main Flow:	
Actor: 1.Customer starts customize settings process. 3.Customer chooses to change the profile settings such as username, mobile number, password and address.	System: 2. System displays interface for customize settings to customer. 4.System allows customer to change any of the chosen option.
Alternative flow:	
Actor: None	System: None
Frequency of use: High	
Technology and Variation: Java (Net beans IDE)	
Special Requirements: **Learn ability:** User interfaces are easy so that customer can understand system functionality.	

9. ID: UC- 09
Use case: Manage Item

Brief Description**:** In this use case manager will manage item in a way that manager can add, update or remove item in system.	
Primary actors: Manager	
Preconditions: There must be some items present in system to update, delete item or add them in system.	
Post Condition: Item changes will be saved in the system	
Main Flow:	
Actor: 1. Manager starts managing item process. 3. Manager update, add, and delete the item in system.	System: 2. System displays interface of manage items. 4. System allows manager to update and add item.
Alternative flow:	
Actor: 3a) Adding Items: Include Add Items	System: None
Frequency of use: High	
Technology and Variation: Java (Net beans IDE)	
Special Requirements: **Learn ability:** System responds to all clicks of manager within 5 seconds.	

10. ID: UC- 10	
Use case: Add Items	
Brief Description**:** In this use case manager will add item id, name, category and price.	
Primary actors: Manager	
Preconditions: Manager should have valid account	
Post Condition: Item will be added in to the system	
Main Flow:	
Actor: 1. Manager chooses to add item such as id, name, category and price. 3. Manager add items in the system.	System: 2. System displays interface of add items. 4. System allows manager to add item.
Alternative flow:	
Actor: None	System: None
Frequency of use: Medium	
Technology and Variation: Java (Net beans IDE)	
Special Requirements: **Learn ability:** System responds to all clicks of manager within 5 seconds.	

11. ID: UC- 11	
Use case: Manage Staff	
Brief Description**:** In this use case manager will manage staff in a way that manager can add, update or remove staff in system.	
Primary actors: Manager	
Preconditions: There must be some staff present in system to update, delete staff or add them in system.	
Post Condition: Staff changes will be saved in the system	
Main Flow:	
Actor: 1. Manager starts managing staff process. 3. Manager update, add, and delete the staff in system.	System: 2. System displays interface of manage staff. 4. System allows manager to update and add staff.
Alternative flow:	
Actor: 3a)Adding Staff : Include Add Staff	System: None
Frequency of use: Medium	
Technology and Variation: Java (Net beans IDE)	
Special Requirements:	

Learn ability: System responds to all clicks of manager within 5 seconds.

12. ID: UC- 12	
Use case: Add Staff	
Brief Description**:** In this use case manager will add staff details such as age, name, salary, past experience.	
Primary actors: Manager	
Preconditions: Manager should have valid account	
Post Condition: Staff will be added in to the system	
Main Flow:	
Actor: 1. Manager chooses to add staff such as age, name, salary, past experience. 3. Manager add staff in the system.	System: 2. System displays interface of add staff. 4. System allows manager to add staff
Alternative flow:	
Actor: None	System: None
Frequency of use: Medium	
Technology and Variation: Java (Net beans IDE)	
Special Requirements:	

Learn ability: System responds to all clicks of manager within 5 seconds.

13. ID: UC- 13	
Use case: Manage Category	
Brief Description**:** In this use case manager will manage category in a way that manager can add, update or delete category in system.	
Primary actors: Manager	
Preconditions: There must be some category present in system to update, delete category or add them in system.	
Post Condition: Category changes will be saved in the system	
Main Flow:	
Actor: 1. Manager starts managing category process. 3. Manager update, add, and delete the categories in the system.	System: 2. System displays interface of manage category. 4. System allows manager to update and add category.
Alternative flow:	
Actor: 3a) Adding Category: Include Add Category	System: None
Frequency of use: Medium	

Technology and Variation: Java (Net beans IDE)
Special Requirements: **Learn ability:** System responds to all clicks of manager within 5 seconds.

14. ID: UC- 14	
Use case: Add Category	
Brief Description**:** In this use case manager will add category details such as id and name.	
Primary actors: Manager	
Preconditions: Manager should have valid account	
Post Condition: Category will be added in to the system	
Main Flow:	
<u>Actor:</u> 1. Manager chooses to add category such as id and name 3. Manager add categories in the system.	<u>System:</u> 2. System displays interface of add category 4. System allows manager to add category
Alternative flow:	
<u>Actor:</u> None	<u>System:</u> None
Frequency of use: Medium	
Technology and Variation:	

Java (Net beans IDE)
Special Requirements: **Learn ability:** System responds to all clicks of manager within 5 seconds.

15. ID: UC- 15	
Use case: Manage Offers	
Brief Description**:** In this use case manager will manage Offer in a way that manager can add, update or remove Offer in system.	
Primary actors: Manager	
Preconditions: There must be some Offers present in system to update, delete Offers or add themin system.	
Post Condition: Offers changes will be saved in the system	
Main Flow:	
Actor: 1. Manager starts managing Offers process. 3. Manager update, add, and delete the Offers in system.	System: 2. System displays interface of manage Offers 4. System allows manager to update and add Offers
Alternative flow:	
Actor:	System:

3a) Adding Offer: Include Add Offer	None
Frequency of use: Medium	
Technology and Variation: Java (Net beans IDE)	
Special Requirements: **Learn ability:** System responds to all clicks of manager within 5 seconds.	

16. ID: UC- 16	
Use case: Add Offers	
Brief Description**:** In this use case manager will add Offers details such as discounts.	
Primary actors: Manager	
Preconditions: Manager should have valid account	
Post Condition: Offers will be added in to the system	
Main Flow:	
<u>Actor:</u> 1. Manager chooses to add Offer such as discounts. 3. Manager add Offers in the system.	<u>System:</u> 2. System displays interface of add Offers 4. System allows manager to add Offers
Alternative flow:	
<u>Actor:</u>	<u>System:</u>

None	None
Frequency of use: Medium	
Technology and Variation: Java (Net beans IDE)	
Special Requirements: **Learn ability:** System responds to all clicks of manager within 5 seconds.	

17. ID: UC- 17	
Use case: Manage Stock	
Brief Description**:** In this use case manager will manage Stock in a way that manager can add, update or delete Stock in system.	
Primary actors: Manager	
Preconditions: There must be some Stock present in system to update, delete category or add themin system.	
Post Condition: Stock changes will be saved in the system	
Main Flow:	
<u>Actor:</u> 1. Manager starts managing Stock process. 3. Manager update, add, and delete the Stock in system.	<u>System:</u> 2. System displays interface of manage Stock 4. System allows manager to update and add Stock.

Alternative flow:	
Actor: 3a) Adding Stock: Include Add Stock	System: None
Frequency of use: Medium	
Technology and Variation: Java (Net beans IDE)	
Special Requirements: **Learn ability:** System responds to all clicks of manager within 5 seconds.	

18. ID: UC- 18	
Use case: Add Stock	
Brief Description**:** In this use case manager will add Stock details such as id and name.	
Primary actors: Manager	
Preconditions: Manager should have valid account	
Post Condition: Stock will be added in to the system	
Main Flow:	
Actor: 1. Manager chooses to add Stock such as id and name 3. Manager add Stock in the system.	System: 2. System displays interface of add Stock 4. System allows manager to add Stock

Alternative flow:	
Actor: None	System: None
Frequency of use: Medium	
Technology and Variation: Java (Net beans IDE)	
Special Requirements: **Learn ability:** System responds to all clicks of manager within 5 seconds.	

19. ID: UC- 19	
Use case: Manage Orders	
Brief Description**:** In this use case manager will manage Orders in a way that manager can accept or reject orders of the customer.	
Primary actors: Manager	
Preconditions: There must be some Orders present in system to reject orders or accept them in system.	
Post Condition: Orders changes will be saved in the system	
Main Flow:	
Actor:	System:

1. Manager starts managing Orders process. 3. Manager accept or reject the Stock in system.	2. System displays interface of manage Orders 4. System allows manager to accept or reject the orders.
Alternative flow:	
Actor: None	System: None
Frequency of use: High	
Technology and Variation: Java (Net beans IDE)	
Special Requirements: **Learn ability:** System responds to all clicks of manager within 5 seconds.	

20. ID: UC- 20	
Use case: Generate bill	
Brief Description**:** In this use case, system will generate bill to the customer after placing order.	
Secondary actors: System	
Preconditions: Customer should place order.	
Post condition: bill will be displayed successfully to the Customer.	
Main Flow:	
Actor: 1. Customer places the order of an item.	System: 2. Displays bill to user.

	3.System saves the information.
Alternative flow:	
Actor: None	System: None
Frequency of Use: High	
Technology and Variations: Java (Net Beans eclipse IDE)	
Special Requirements: **User Interface:** System displays a user-friendly interface so that user can understand it easily.	

Artifact #4
Activity Diagram

Customer:

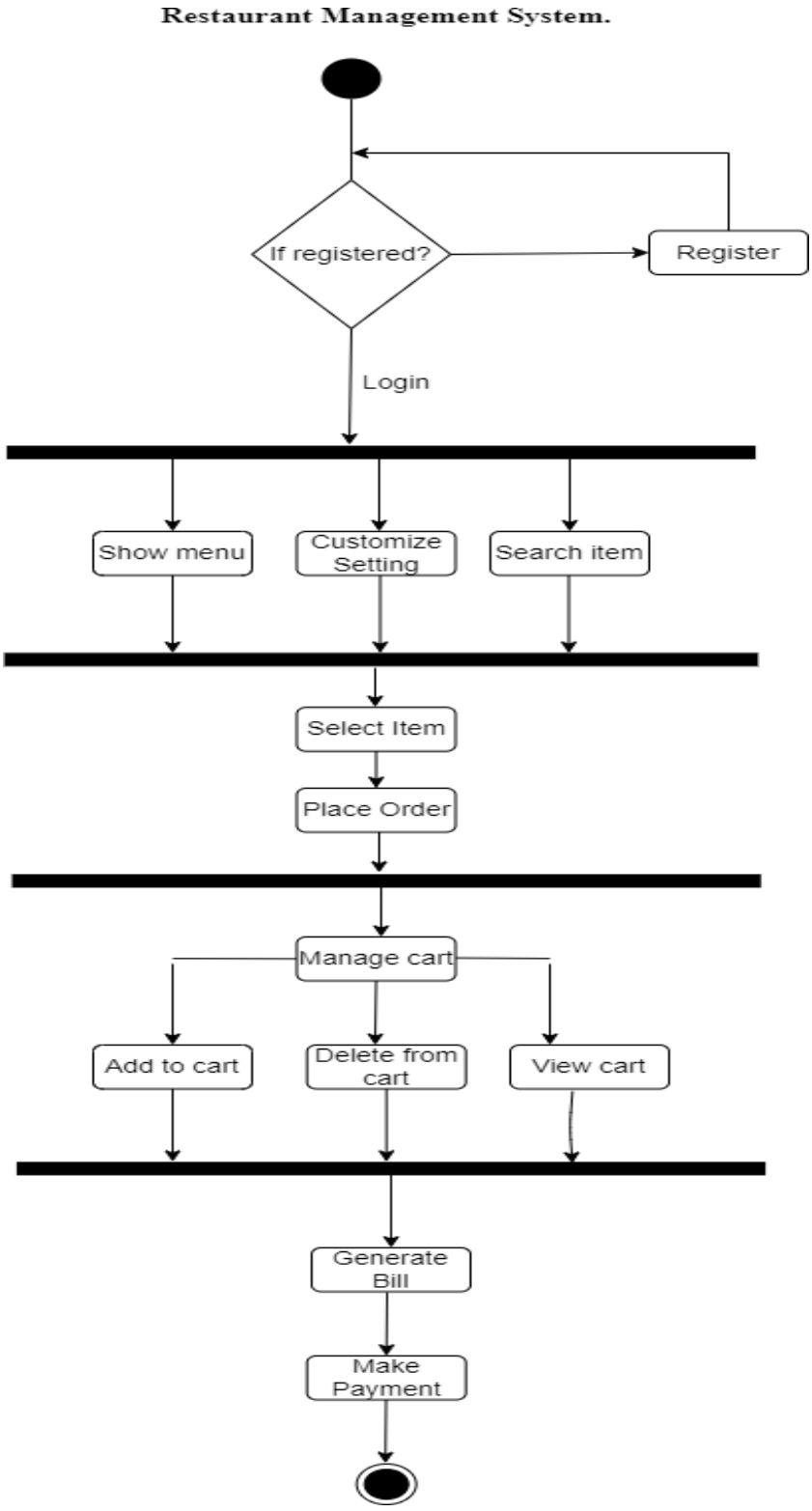


Fig 4 Activity Diagram for Customer of Café Management System

Manager:

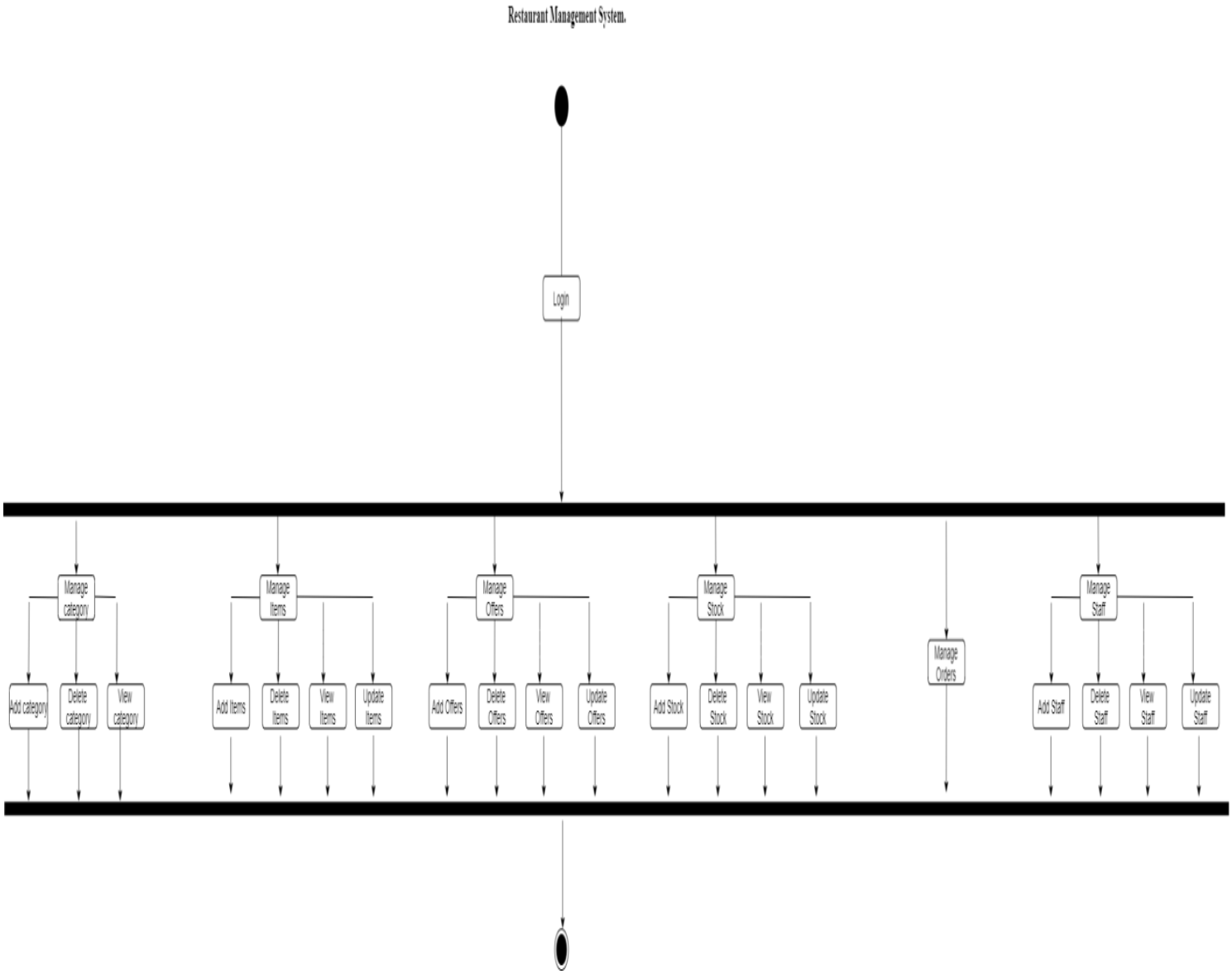


Fig 4.1 Activity Diagram for Manager of Café Management System

Artifact # 5
Domain Model

Domain Model

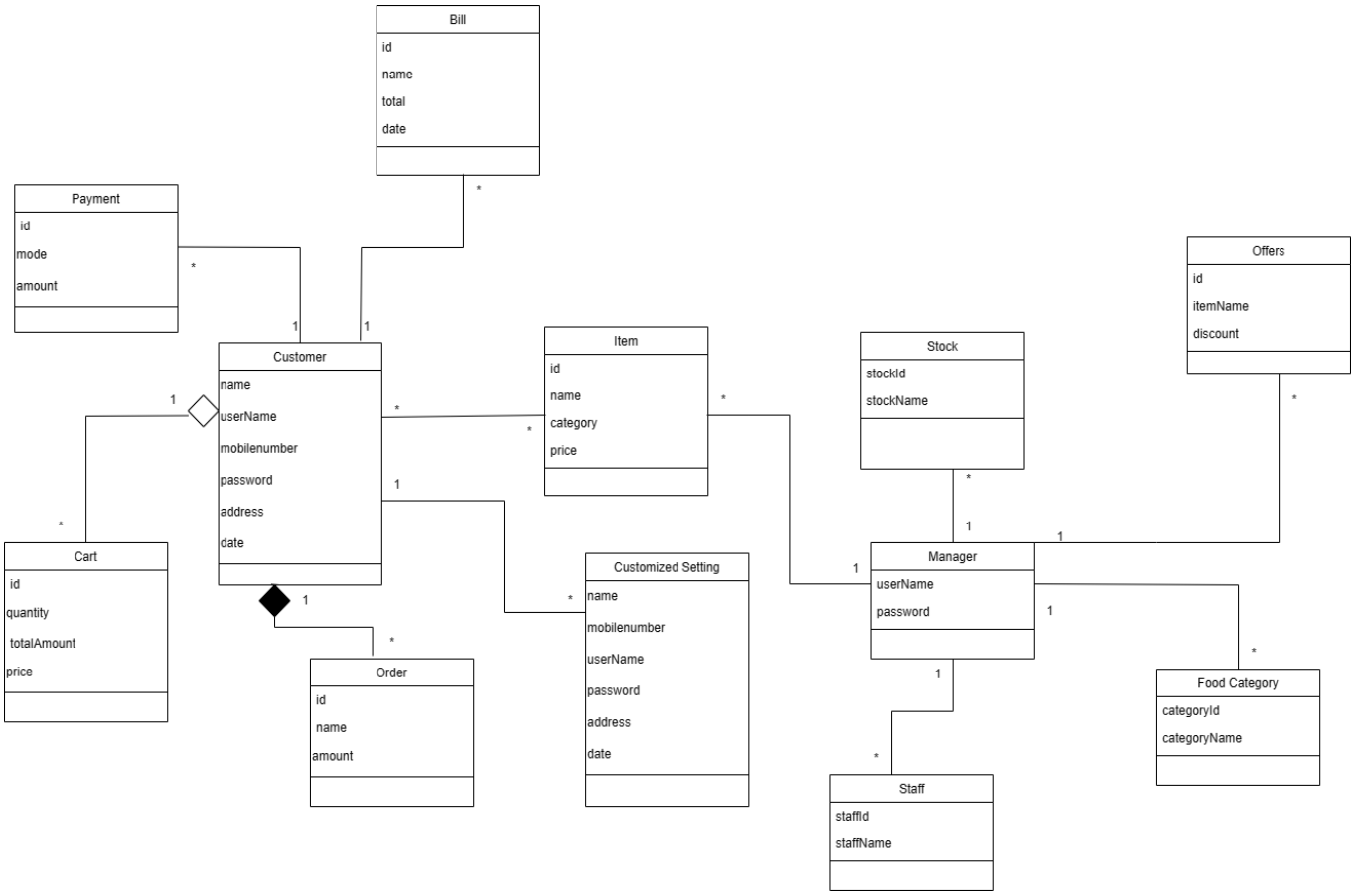
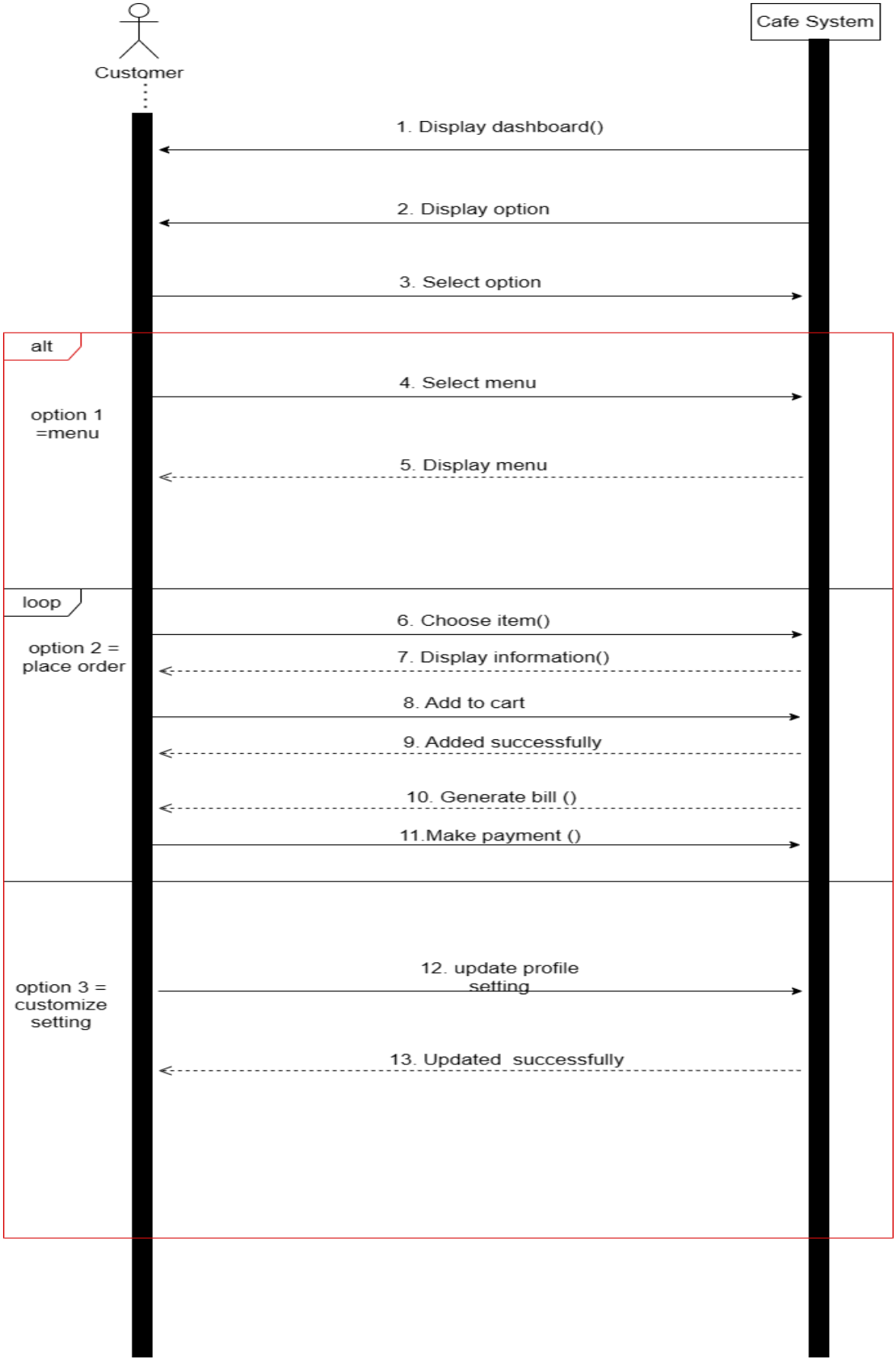


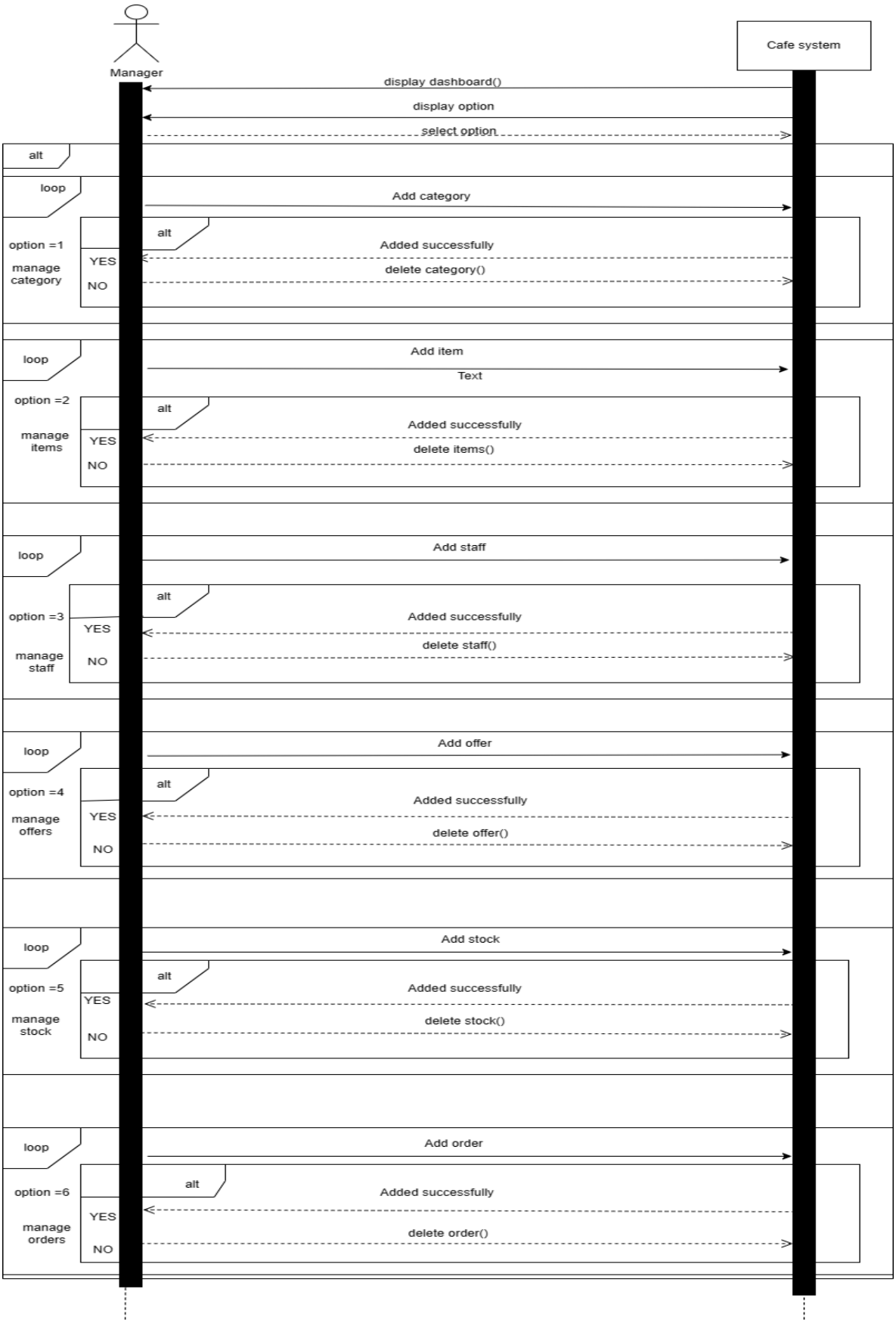
Fig 5 Domain Model

Artifact # 6
Sequence Diagram

Customer:


Customer
Cafe System
1. Display dashboard()
2. Display option
3. Select option
alt
option 1 =menu
4. Select menu
5. Display menu
loop
option 2 = place order
6. Choose item()
7. Display information()
8. Add to cart
9. Added successfully
10. Generate bill ()
11.Make payment ()
option 3 = customize setting
12. update profile setting
13. Updated successfully

Manager:


Manager
Cafe system
display dashboard()
display option
select option
alt
loop
Add category
option =1
manage category
alt
YES
NO
Added successfully
delete category()
loop
Add item
Text
option =2
manage items
alt
YES
NO
Added successfully
delete items()
loop
Add staff
option =3
manage staff
alt
YES
NO
Added successfully
delete staff()
loop
Add offer
option =4
manage offers
alt
YES
NO
Added successfully
delete offer()
loop
Add stock
option =5
manage stock
alt
YES
NO
Added successfully
delete stock()
loop
Add order
option =6
manage orders
alt
YES
NO
Added successfully
delete order()

Artifact #7
Class Diagram

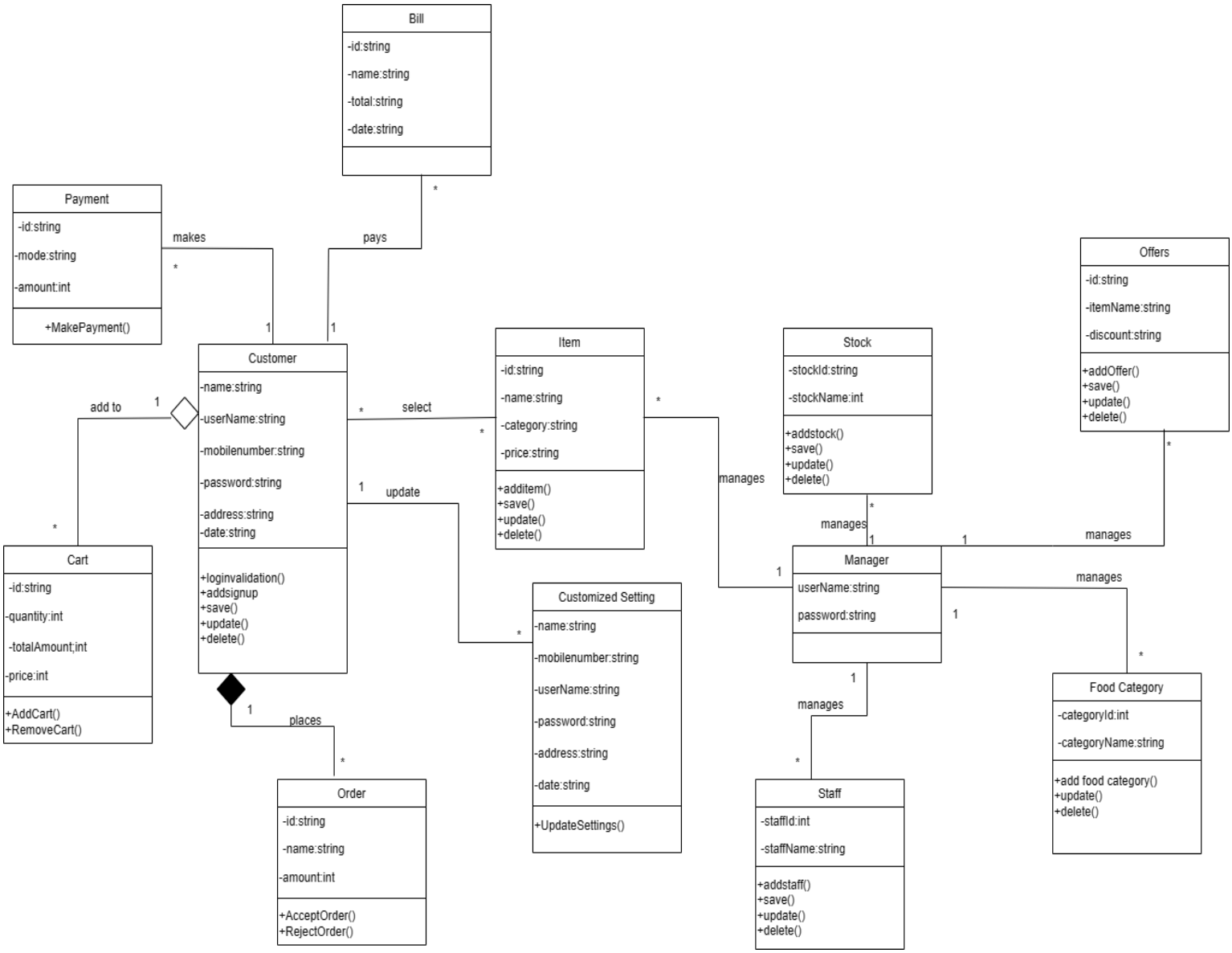


Fig 7 Class Diagram

Artifact #8
State Transition Diagram

Manager

Manage category:

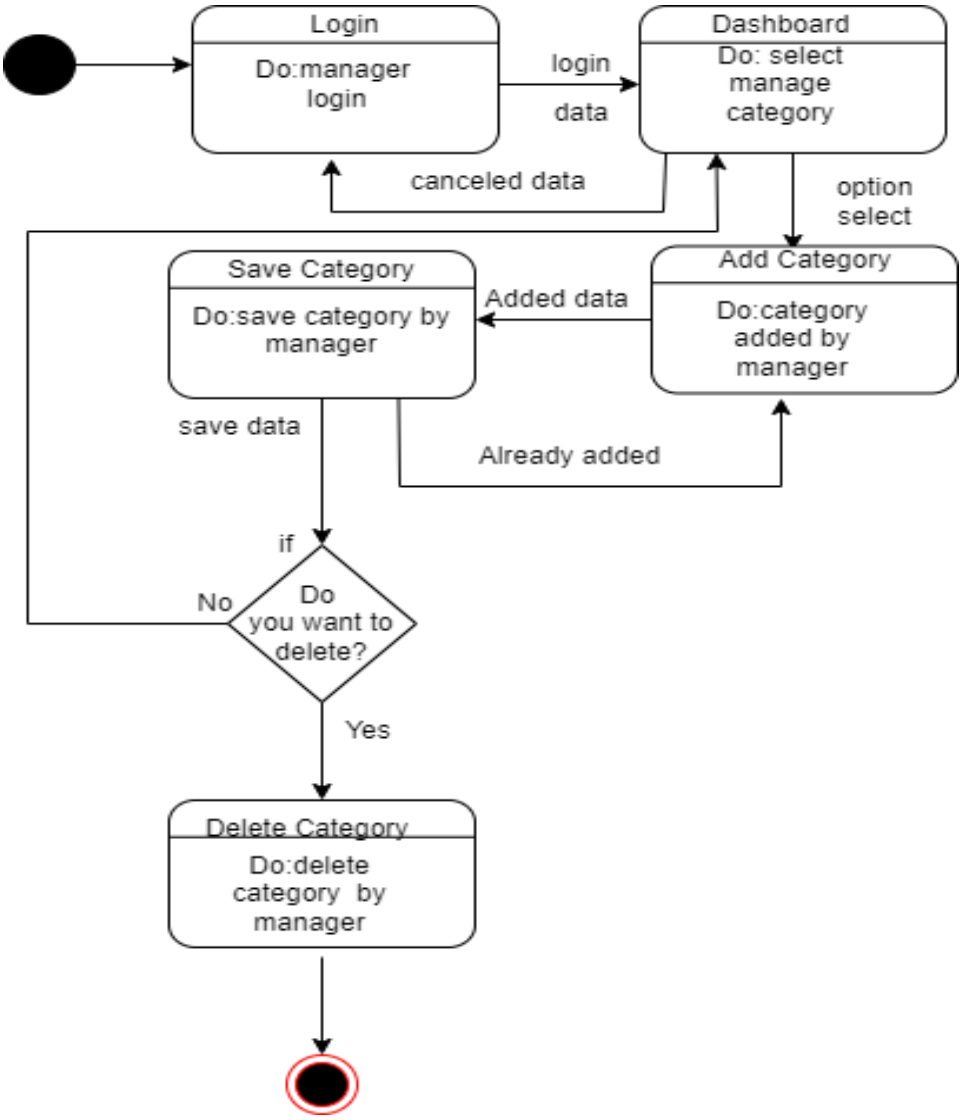


Fig 8 State Transition Diagram for Manager of Café Management System

Manage Items:

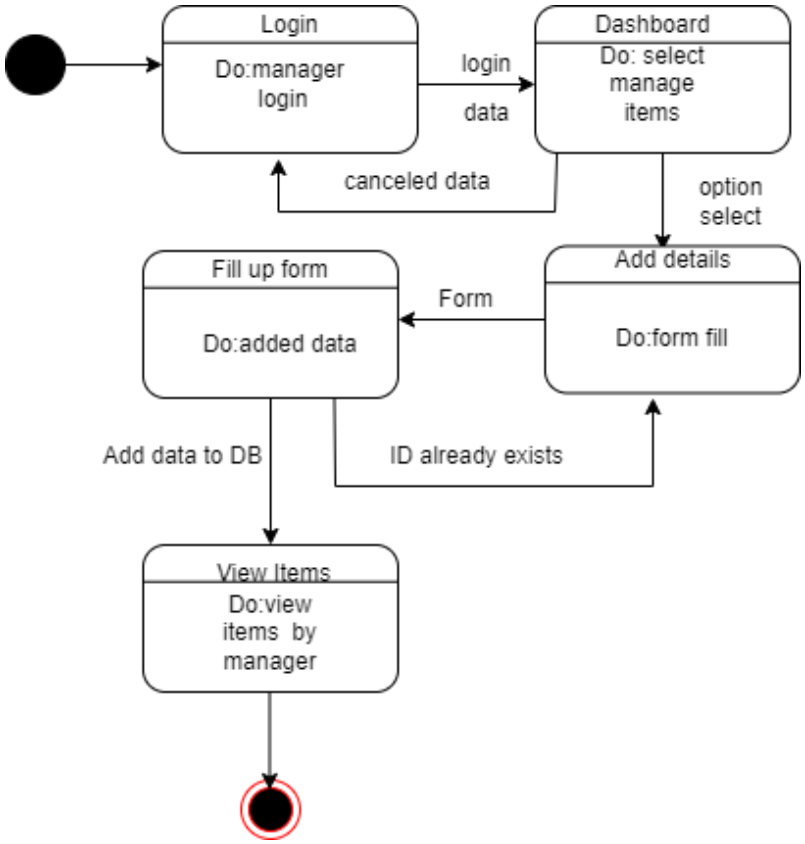


Fig 8.1 State Transition Diagram for Manager of Café Management System

Manage Staff:

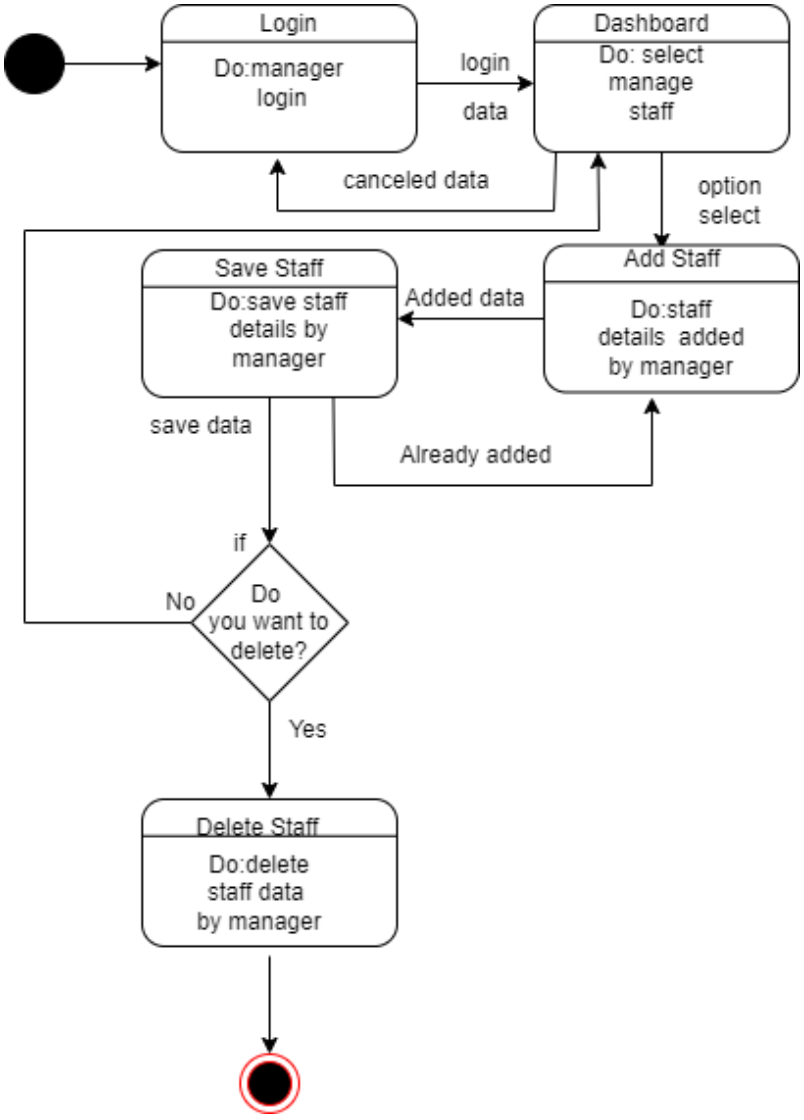


Fig 8.2 State Transition Diagram for Manager of Café Management System

Manage Stock:

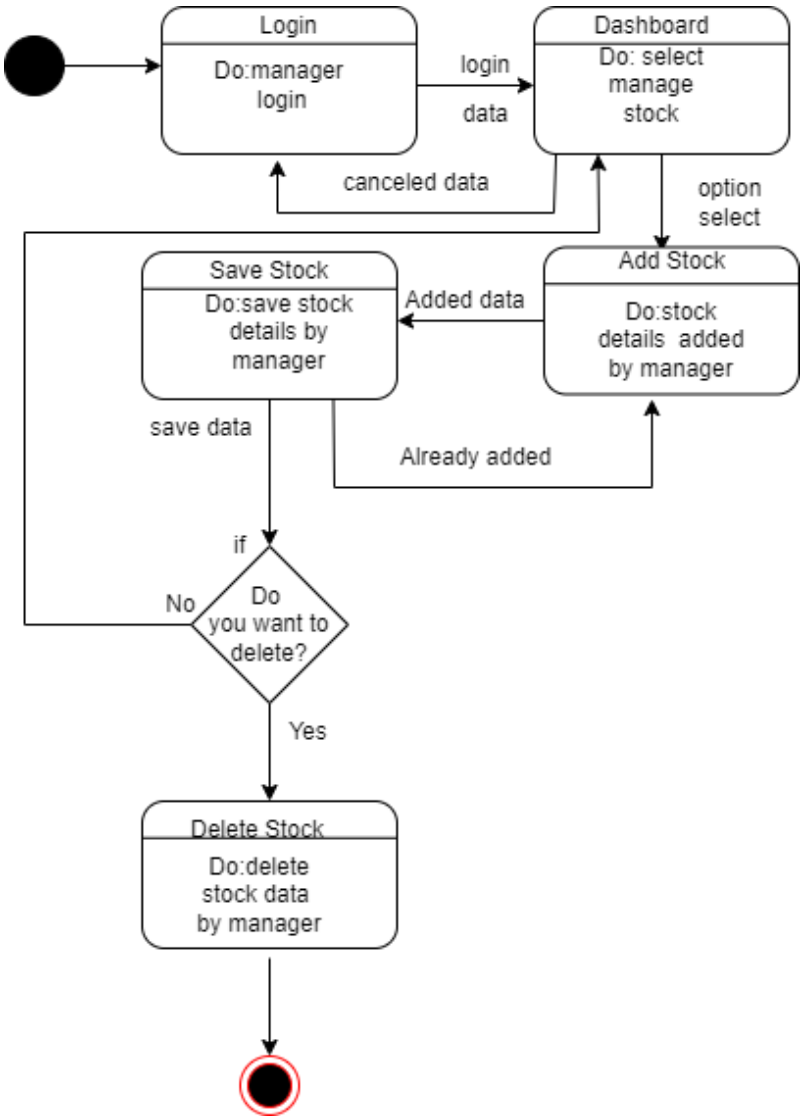


Fig 8.3 State Transition Diagram for Manager of Café Management System

Customer

Place Order:

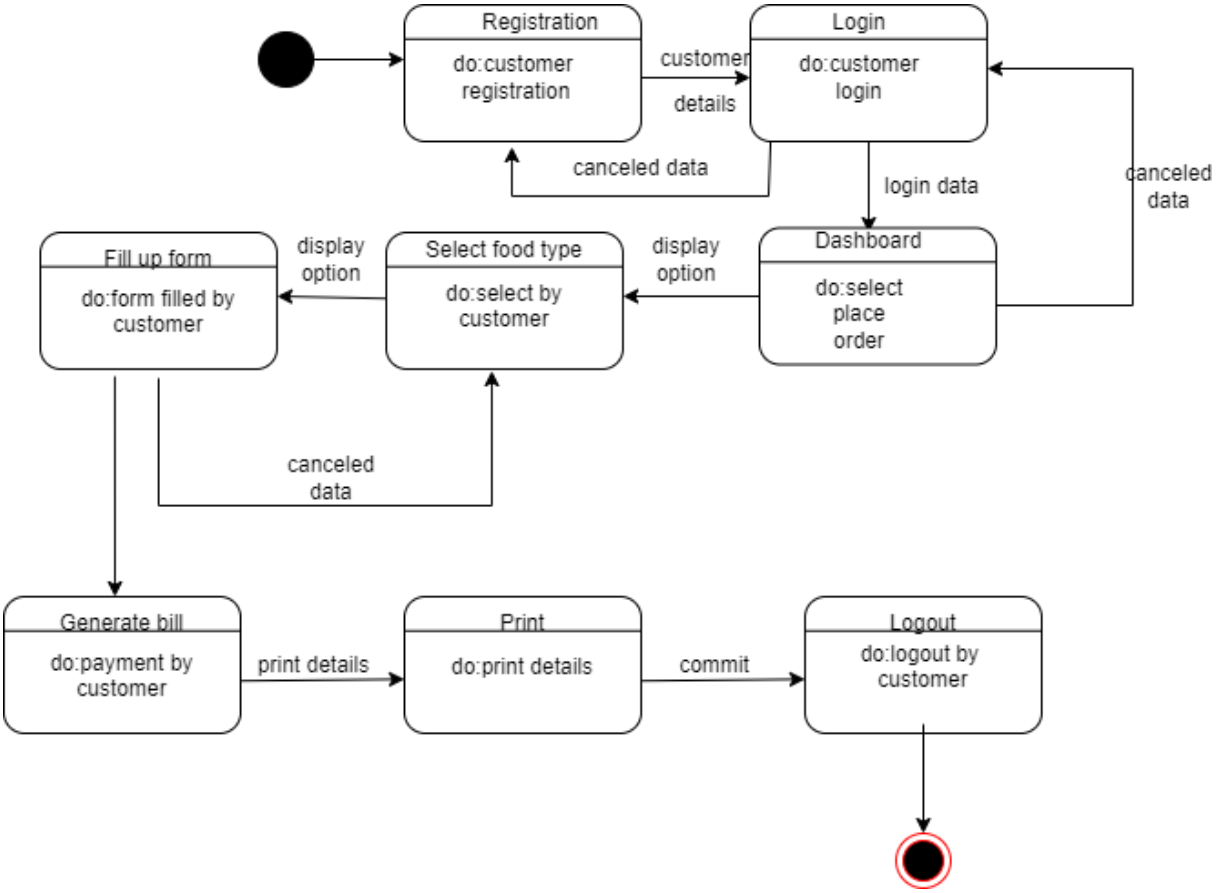


Fig 8.4 State Transition Diagram for Customer of Café Management System

Menu:

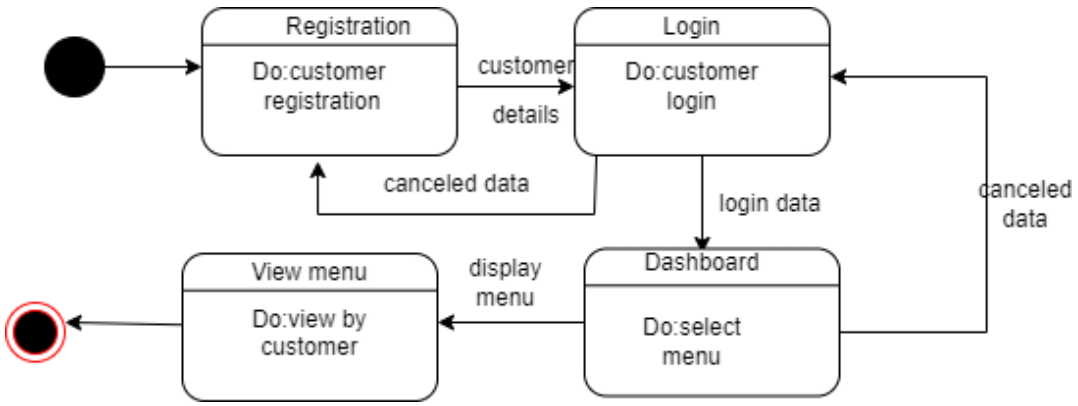


Fig 8.1 State Transition Diagram for Customer of Café Management System